Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Rok Jelovšek

Navidezna kabina za pomerjanje oblek na spletu

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Matija Marolt

Ljubljana, 2017

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomski nalogi izdelajte navidezno kabino za pomerjanje oblačil na spletu. Raziščite področje simulacije tkanin in izdelajte aplikacijo, ki bo delovala na podlagi vnaprej pripravljenih 3D modelov, na katere najprej poravna oblačilo, nato pa s simulacijo prikaže kako se oblačilo modelu prilega.

Zahvaljujem se mentorju doc. dr. Matiji Maroltu za priložnost in pomoč pri izbiri teme diplomske naloge. Prav tako bi se rad zahvalil asistentu dr. Cirilu Bohaku, da me je uspešno usmerjal pri izdelavi diplomske naloge. Zahvalil bi se tudi svoji družini, ki mi je pri izdelavi naloge stala ob strani in me spodbujala.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
RGB	red, green, blue	rdeča, modra, zelena
HTML	Hyper Text Markup Language	označevalni jezik za izdelavo spletnih strani
XML	Extensible Markup Language	razširljiv označevalni jezik
3D	three-dimensional	tridimenzionalno
WebGL	Web Graphics Library	knjižnica za strojno pospešen izris grafike na spletu
2D	two-dimensional	dvodimenzionalno
DB	data base	podatkovna baza
AJAX	asynchronous JavaScript and XML	asinhroni JavaScript in XML
CSS	Cascading Style Sheets	kaskadne stilske podloge
SVG	Scalable Vector Graphics	umerljiva vektorska grafika
OBJ	.obj file	.obj datoteka
MTL	.mtl file	.mtl datoteka

Povzetek

Naslov: Navidezna kabina za pomerjanje oblek na spletu

Avtor: Rok Jelovšek

S tem delom želimo prikazati simulacijo tkanine, jo prenesti na 3D modele oblačil in jih uporabiti pri oblačenju 3D modela človeka. Za simulacijo uporabljamo metodo mase in vzmeti, ki tkanino razdeli na delce določene mase in jih poveže z različnimi vzmetmi. Premikanje delcev se vrti okoli Newtonovega drugega zakona, z vzmetmi pa poskušamo preprečiti deformacije. Za spreminjanje 3D modela človeka uporabljamo izpeljanko prostorske metode deformacije, za izrisovanje na spletni strani knjižnico Three.js. Za detekcijo trkov med modelom oblačila in človeka uporabljamo prostorsko razdelitev. Kot obiskovalci lahko prilagodimo model človeka, da nam je kar se da podoben, izberemo spol, telesne mere, ponujen kos oblačila, velikost in barvo. Odločimo se lahko med mehkim in trdim materialom, kar aplikacijo naredi bolj fleksibilno. Nato lahko zaženemo simulacijo in si ogledamo, kako se oblačilo prilagaja telesu.

Ključne besede: simulacija, tkanina, model, oblačenje, splet, knjižnica, delec, vzmet.

Abstract

Title: Virtual online fitting-room

Author: Rok Jelovšek

The aim was to familiarize ourselves with cloth simulation, translate it into 3D clothing models and use them to dress a 3D human model. For the simulation, a mass-spring model was used, which divides the cloth into many particles with mass and connects them by various springs. Particle movement is based on Newton's second law and springs have been put in place to prevent deformations. An extension of the space deformation method and the Three.js library for drawing on a website were used to change the 3D human model. For detecting collision between the clothing and the human model, a space division method was used. As visitors we can modify the human body model, so that it looks like us as much as possible; choose its gender, body sizes, clothing, clothing size and color. We can choose between a soft and hard material, which makes the application more flexible. Then we can start the simulation and see how the clothing conforms to the body.

Keywords: simulation, cloth, model, dressing, web, library, particle, spring.

Poglavje 1

Uvod

Ljudje vedno več časa preživljamo v navideznem okolju in rezultate hočemo vedno hitreje. Tudi spletno nakupovanje postaja vedno bolj popularno, saj iz udobja domačega stola lahko kupimo praktično karkoli. Vendar smo ljudje, ko pride do nakupovanja oblačil, bolj občutljivi, saj jih pred nakupom želimo tudi pomeriti, da se na lastne oči prepričamo, kako majica izgleda na nas. Pričujoča diplomska naloga ne rešuje spletnega nakupovanja, ampak poskuša reševati pomerjanje oblačil v navideznem okolju. Ta problem seveda nastopi tudi na drugih področjih; najhitreje ga opazimo že ob igranju nekaterih računalniških iger. V teh igrah običajno najprej s pomočjo urejevalnika telesa izdelamo našega junaka, nato pa zanj lahko izberemo še razna oblačila. Namesto, da bi imeli izdelovalci iger pripravljenih tisoče modelov človeka z različnimi oblačili, jih imajo lahko le kakšnih deset, na njih pa lahko pomerimo oblačila. Tako so lahko modeli junaka bolj kvalitetno narejeni, saj jih ne potrebujemo toliko. V drugih računalniških igrah želimo le, da je obnašanje oblačil podobno kot v realnem življenju, da npr. krilo popleše, ko vanj zapiha veter ipd. Želimo tudi, da pomerjanje oblačil na spletu izgleda čimbolj realno. Tu pa pridemo do glavne teme te diplomske naloge simulacije tkanin.

1.1 Cilji

Glavna cilja diplomske naloge sta simulacija tkanin in preko nje prilagajanje oblačila 3D modelu človeškega telesa na spletu v realnem času. Pri tem seveda želimo, da je obnašanje modela obleke čim bolj podobno pravi tkanini. Če želimo, da si uporabnik zna predstavljati, kako oblačilo izgleda na njegovem telesu, je eden izmed ciljev pred njega postaviti parametriziran 3D model človeka. Tako lahko prilagodi obliko modela in dobi predstavo o tem, kako bi oblačilo izgledalo na njegovem telesu v realnem življenju. Da je vse to mogoče, se je najprej potrebno seznaniti s knjižnico Three.js, ki omogoča grafični prikaz na spletu. Ko nam uspe pridobiti in na splet postaviti parametriziran 3D model človeka, potrebujemo samo še modele oblačil, ki jih bomo ponudili uporabnikom, nato pa lahko začnemo s pisanjem simulacijske kode.

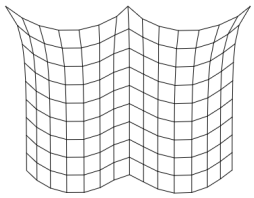
1.2 Struktura dela

Diplomsko delo je sestavljeno iz petih poglavij. V drugem poglavju preverimo metode simulacije tkanin. V istem poglavju pregledamo 3D model, parametriziran 3D model človeka in na koncu preverimo obstoječe rešitve. V tretjem poglavju govorimo o uporabljenih orodjih in tehnologijah. Najprej predstavimo programski jezik, tehnologije spletne strani in uporabljene knjižnice. Na koncu poglavja predstavimo program Blender, format OBJ in MTL. V četrtem poglavju predstavimo algoritem glavnega programa, v podpoglavjih pa glavne metode. Peto poglavje je zaključek, v katerem povzamemo rezultate in težave, ter navedemo ideje za izboljšave.

Poglavje 2

Pregled področja

Pri simulaciji tkanin se največkrat srečamo z delci, ki imajo svojo maso in so med seboj povezani na različne načine. S temi vezmi običajno tvorijo mrežo, kot je prikazano na sliki 2.1.



Slika 2.1: Primer mreže delcev.

Za določanje obnašanja vsakega delca se v simulacijski kodi oz. fizikalnem pogonu uporablja drugi Newtonov zakon (enačba 2.1). Zakon določa, da je sila produkt mase in pospeška. S tem poskušamo izračunati položaj delcev in dobiti obliko tkanine. Ena izmed metod, ki se lahko uporablja za izračun smeri potovanja delcev zaradi sil, je Verletova integracija.

$$\vec{F} = m\vec{a} \tag{2.1}$$

Prvi interesi za računalniško simulacijo tkanin so se pojavili že v prejšnjem stoletju okoli leta 1986, ko je izšlo več člankov na temo različnih metod. Poznamo dve glavni: fizikalna in geometrijska metoda.

2.1 Geometrijske metode

Leta 1986 je Weil [18] predstavil metodo za modeliranje tkanine, ki je sestavljena iz fiksnega števila točk. Tkanino je smatral kot skupino kablov. Pri tej metodi je modelirana kot 2D mreža 3D točk in se računa v dveh korakih:

- Točke rekurzivno povežemo s krivuljami v obliki hiperboličnega kosinusa (verižnica na sliki 2.2) in tiste, ki visijo zunaj konveksne ovojnice kabla, odstranimo.
- Drugi korak je sproščanje. Ta korak je pomemben, saj v njem zagotovimo, da se delci držijo razdalje, ki smo jo definirali (razdalja med točkami, ki so povezane s krivuljo). Rezultat je gladka tkanina z realnimi pregibi.

Slika 2.2: Viseča veriga ima obliko krivulje, ki ji pravimo verižnica.

2.2 Fizikalne metode

Te metode tkanino obravnavajo kot mrežo med seboj z vzmetmi povezanih delcev. Poznamo tri glavne tipe metod:

- metode raztegljivosti,
- metode delcev,
- metode mase in vzmeti.

2.2.1 Metode raztegljivosti

To tehniko so prvič objavili leta 1986, ravno ob izdaji Weilovega članka o geometrijskih metodah[14]. Tehnika definira energijske funkcije skozi 2D mrežo 3D točk. Energija modela je opisana kot raztezek, togost in teža (enačba 2.2).

$$E = k_s E_s + k_b E_b + k_g E_g \tag{2.2}$$

- E_s predstavlja energijo kot posledico raztezanja (z upoštevanjem Hookovega zakona, enačba 2.4).
- E_b predstavlja energijo kot posledico ukrivljanja.
- E_g predstavlja energijo kot posledico gravitacije (težni pospešek).
- Drugi členi so faktorji materiala.

Ideja metode je, da tkanino obravnavamo kot material, ki zaseda celoten prostor, kjer se nahaja. Če pogledamo v kozarec vode, vidimo, da je poln celoten kozarec, vendar vemo, da je voda zgrajena iz atomov, ki so med seboj razmaknjeni. Pri tej metodi to ignoriramo in predpostavimo, da tkanina zajema celoten prostor. Razdalja med točkami in meritev ukrivljenosti te razdalje sta vse, kar potrebujemo za izračun sil elastičnosti v tkanini.

2.2.2 Metode delcev

Leta 1992 sta Breen in House predstavila metodo, osnovano okoli delcev[12], ki metodo raztegljivosti pelje še dlje. Razvila sta model za simulacijo tkanin z uporabo interaktivnega sistema delcev, ki predstavlja mehanično strukturo tkanine. Delci interaktirajo s sosednjimi delci na podlagi enačb, ki definirajo mehanične povezave, predstavljene z energijskimi funkcijami. S pomočjo energetskih interakcij med delci lahko določimo obliko tkanine.

$$U_T = U_R + U_S + U_B + U_T + U_G \tag{2.3}$$

V enačbi 2.3 vidimo vsoto energij U_T, kjer U_R predstavlja izmišljen element, ki ga dodamo, da preprečimo križanje delcev. U_S je energija raztezanja tkanine (Hookov zakon), U_B je energija, ki opisuje togost, U_T je energija, ki opisuje deformacijo tkanine, v kateri vzporedne notranje površine zdrsnejo druga čez drugo, U_G pa je energija, ki nastopi kot posledica težnega pospeška.

2.2.3 Metode mase in vzmeti

To tehniko sta leta 1988 prvič predstavila Haumann in Parent[13], naprej pa jo je razvil Provot leta 1995[17] in je uporabljena tudi v praktičnem delu te diplomske naloge. Tkanina je v tej metodi predstavljena kot mreža m x n točk. Vsaka točka ima svojo maso, med seboj pa so povezane z vzmetmi brez nje. Na začetku simulacije, v fazi inicializacije, se kot začetna razdalja med točkama zabeleži dolžina vsake vzmeti, ki ju ta vzmet povezuje. Poznamo več vrst vzmeti, ki jih ločimo glede na to, katera delca povezuje in kakšen je njihov namen.

- Vzmetem, ki povezujejo točko $[i, j]$ z $[i+1, j]$, in $[i, j]$ z $[i, j+1]$, pravimo **strukturne** vzmeti (angl. structural springs). Kot lahko razberemo iz izrazov, ta vrsta povezuje delce, ki so si v x smeri horizontalni, v y smeri vertikalni in v obeh smereh sosednji.

- Vzmetem, ki povezujejo točko $[i, j]$ z $[i+1, j+1]$, in $[i+1, j]$ z $[i, j+1]$, pravimo **strižne** vzmeti (angl. shear springs). Kot lahko razberemo iz izrazov, ta vrsta povezuje delce, ki so si med seboj diagonalni v x in y smeri ter diagonalno sosednji.

- Vzmetem ki povezujejo točko $[i, j]$ z $[i+2, j]$, in $[i, j]$ z $[i, j+2]$, pravimo **upogibne** vzmeti (angl. flexion/bending springs). Kot lahko razberemo iz izrazov, ta vrsta povezuje delce, ki so si v x smeri horizontalni in v y smeri vertikalni. Ne povezuje sosednjih delcev, ampak delec z delcem, ki je takoj za sosednjim.

(a) Strukturne vzmeti (b) Strižne vzmeti (c) Upogibne vzmeti

Slika 2.3: Tipi vzmeti.

Vsaka vrsta ima svojo vlogo. Strukturne vzmeti so namenjene za upor proti raztezanju. Strižne vzporednim površinam tkanine preprečujejo, da ne zdrsnejo druga preko druge in ostanejo poravnane. Upogibne vzmeti so namenjene za upor proti gubanju. Ko so vse vzmeti in delci pripravljeni, se simulacija nadaljuje z integracijo Newtonovega drugega zakona (enačba 2.1). Ko na delce tkanine delujejo zunanje sile, npr. gravitacija, se to zgodi v nekem časovnem intervalu. Iz Newtonovega drugega zakona lahko izračunamo pospešek, saj je masa delca znana. Tako lahko s pomočjo časovnega koraka, ki je znan ob vsakem zagonu simulacije, izračunamo tudi hitrost vsakega delca. Posledica hitrosti je premik in s tem podaljšanje/krčenje vzmeti med delci. Tukaj nastopijo notranje sile vzmeti in lahko uporabimo Hookov zakon (enačba 2.4), ki velja, dokler sile ne prekoračijo meje sorazmernosti oz. meje elastičnosti. Zakon velja za širjenje in krčenje vzmeti, saj je v primeru krčenja tudi sila negativna.

$$F = k(l_2 - l_1) \tag{2.4}$$

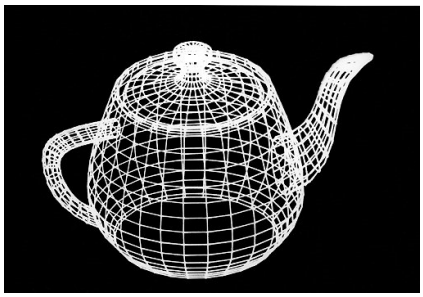
Hookov zakon: Sila F je produkt materiala vzmeti k (elastičnost) in spremembe dolžine vzmeti $(l_2 - l_1)$. Da gravitacija v simulaciji vzmeti ne razteguje predolgo, je prisoten tudi **faktor blaženja** (angl. damping).

$$F_b = -C_b v_i \tag{2.5}$$

Sila blaženja F_b je produkt faktorja blaženja C_b in hitrosti delca v_i. Vsota sil, ki delujejo na delec in vzmet, je integrirana glede na časovni interval in rezultat je pospešek na vsakem delcu.

2.3 3D model

3D modeli predstavljajo fizična telesa, z uporabo točk v 3D prostoru, ki ponazarjajo njihovo geometrijo. Točke so povezane z raznimi geometrijskimi oblikami, kot so npr. trikotniki, štirikotniki, črte, krivulje itd. Lahko jih izdelamo ročno (točke definiramo sami ali s pomočjo risalnega programa), algoritmično (npr. v programu, kot je Blender, lahko naredimo škatlo ali druga telesa poljubnih velikosti, ki se generirajo avtomatično z definiranim algoritmom) ali z uporabo 3D optičnih čitalnikov. Njihova površina je lahko vnaprej definirana z raznimi materiali in teksturami. Uporabljajo se na raznih področjih, npr. v medicini, filmski industriji itd.



Slika 2.4: Vizualizacija geometrije posode, točke so povezane v štirikotnike.

2.4 Parametriziran 3D model človeka

Slika 2.5: Parametriziran 3D model človeka.

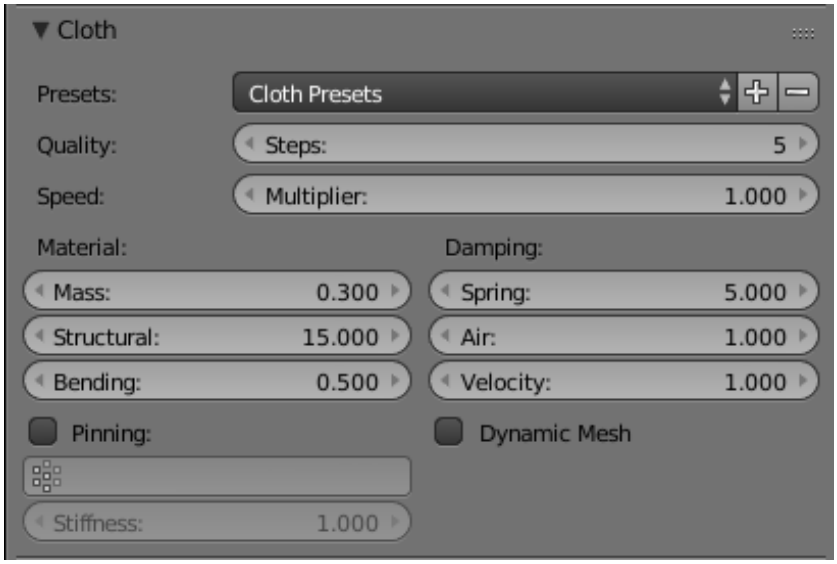
Ena izmed novejših tehnik generiranja 3D človeških modelov je uporaba 3D optičnih bralnikov. Tako dobimo na milimeter natančen 3D model, ven-

dar ima takšen model preveč informacij, zato procesiranje traja predolgo in ni primeren za aplikacije, ki se izvajajo v realnem času. Ker hočemo večjo učinkovitost, model raje naredimo manj realen. Prvi korak pri ustvarjanju parametriziranega 3D modela človeka je oblikovanje začetnega človeškega modela (angl. template) in določanje točk delov telesa, ki bodo nastopile v posamezni deformaciji (razdelimo jih na regije). Deformacija je proces, ki spremeni točke telesa. Definiranje regij je potrebno izvesti ročno, avtomatiziran proces nam model lahko razdeli samo po glavnih okončinah ali približno. Deformacija neke regije, ki smo jo na začetku določili, je dosežena šele, ko sprememba vpliva na dve regiji ali več. Nato potrebujemo še funkcijo, ki zgladi spremembe vseh prekrivajočih se regij. Ena izmed metod je prostorska deformacija (angl. space deformation), kjer nekatere točke 3D modela premaknemo iz enega prostora v drugega[16]. Obstaja pa mnogo izpeljank te metode in vse delujejo drugače. Nekatere model obdajo z geometrijskim telesom, ki ga deformirajo, poleg tega pa deformirajo tudi 3D model človeka. Obstaja tudi metoda, kjer model razrežemo na rezine in jim spreminjamo radij.

2.5 Obstoječe rešitve

2.5.1 Simulacije tkanine

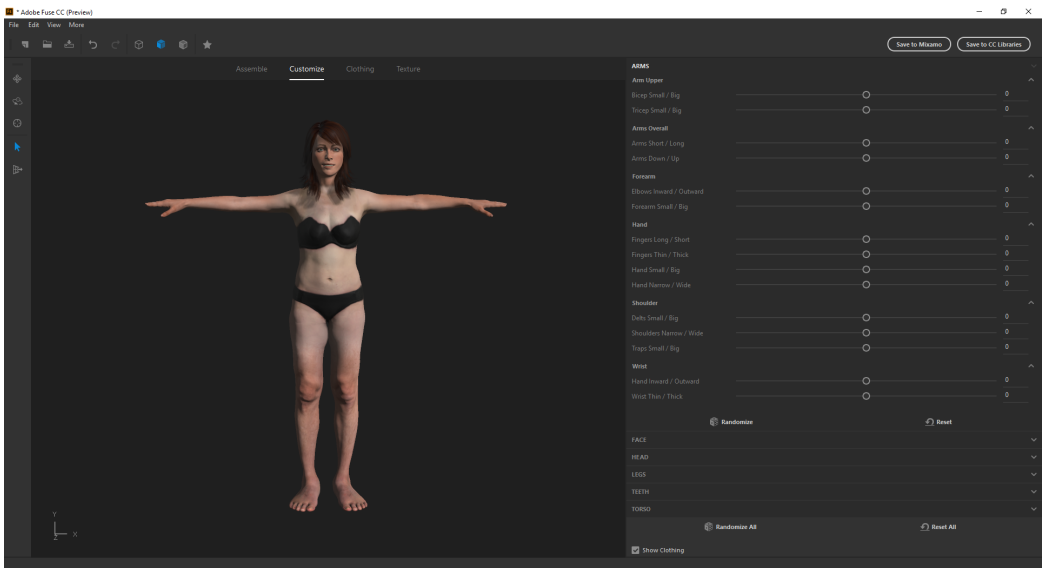
Simulacija tkanin je danes že praktično rešen problem. Simuliramo jo lahko v programu Blender[1], ki uporablja model mase in vzmeti. Kot vidimo na sliki 2.6, lahko nastavimo razne parametre materiala, npr. kakšna je masa delcev in kako močne so omejitve, ki jih definirajo vse tri vrste vzmeti. Definiramo lahko tudi faktor vzmeti, upor zraka in faktor blaženja. Uporabljamo jo lahko tudi v drugih večjih programih, kot sta 3ds Max [3] in Maya[4]. Na spletu lahko najdemo tudi stran The Cloth Simulation[15], ki jo je naredil Andrew Hoyer. Zopet gre za simulacijo po metodi mase in vzmeti, vendar v tem primeru zgolj 2D mreže točk.



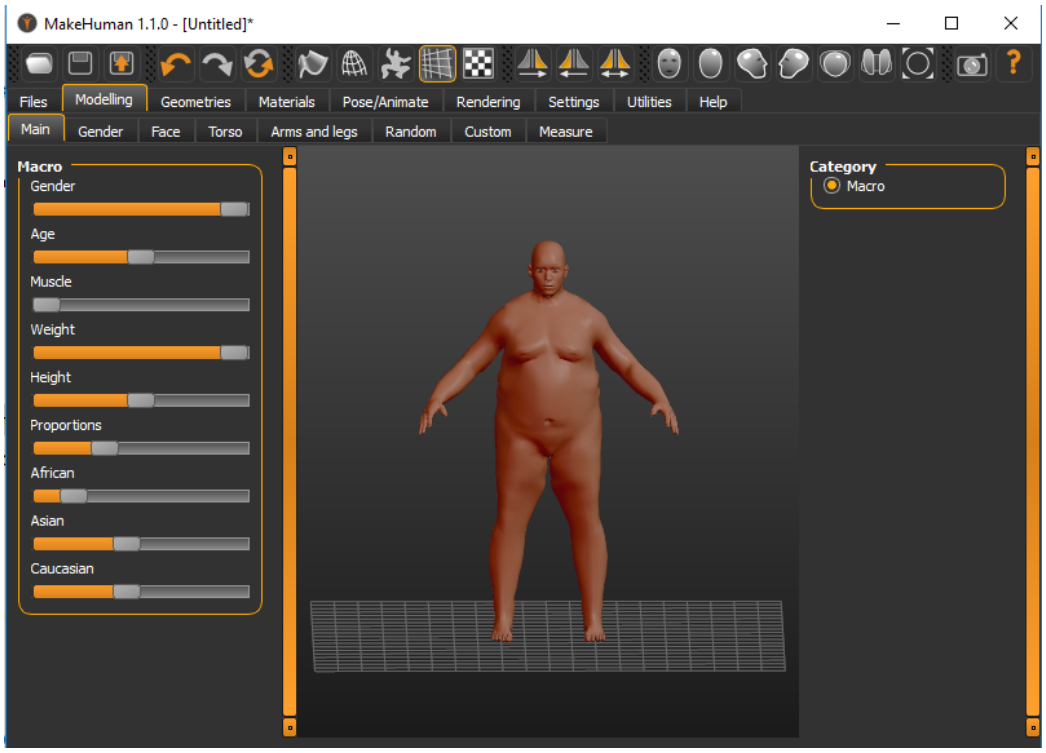
Slika 2.6: Simulacija tkanin v programu Blender.

2.5.2 Parametriziran 3D model človeka

Za pridobitev parametriziranega modela človeka lahko uporabimo Adobe Fuse CC[10]. V tej aplikaciji lahko ustvarimo poljuben model človeka, konstrukcijo začnemo z izbiranjem trupa, nog, rok in glave. Nadaljujemo lahko s spreminjanjem dimenzij delov telesa in gremo lahko še v večje podrobnosti, kot je usmerjenost obrvi ali velikost zob. Nastavimo lahko skoraj vse zunanje lastnosti telesa, pomerimo lahko razna oblačila (hlače, majico, čevlje itd.) in model izvozimo v formatu OBJ. Podobno lahko naredimo v odprtokodni aplikaciji MakeHuman[6], le da imamo na voljo manj opcij, ki je napisana v programskem jeziku Python, vendar pa gre za ogromen projekt in dobiti kodo iz tako velikega projekta je praktično nemogoče, saj bi bilo potrebno proučiti celotno kodo (slaba javna dokumentacija). Na spletu lahko najdemo tudi Body Visualizer[2]. Na tej spletni strani je implementiran parametriziran 3D model človeka. Celotna koda je napisana v programskem jeziku JavaScript in uporablja ogrodje WebGL.



Slika 2.7: Program Adobe Fuse CC s parametriziranim 3D modelom človeka.



Slika 2.8: Program MakeHuman s parametriziranim 3D modelom človeka.

Poglavje 3

Pregled tehnologij in orodij

Pri izdelavi diplomske naloge je bil uporabljen programski jezik JavaScript[5]. To je jezik, s katerim spletne strani lahko naredimo interaktivne in danes velja za enega obveznih predznanj vsakega spletnega razvijalca. Program se sproti interpretira na uporabnikovem računalniku, zato za strežnike ni obremenjujoč. Podpora za programski jezik JavaScript je že vgrajena v vse večje spletne brskalnike. Podpira objektno, imperativno in funkcionalno naravnano programiranje. Je zelo odprt jezik in ne potrebuje veliko znanja sintakse, saj tudi ne uporablja tipov spremenljivk. Hitro se ga lahko naučimo na domači spletni strani ali še lažje na brezplačni strani W3Schools[11]. JavaScript se uporablja tudi na drugih področjih (v PDF dokumentih, MongoDB itd.), ne samo v HTML (Hyper Text Markup Language) dokumentih. V kodi na naslednji strani lahko vidimo, kako izgleda JavaScript program v HTML dokumentu. Navadno HTML spletno stran, ki jo prikažejo brskalniki, naredi dinamično. V programu dodamo odziv na klik gumba Hello in ob kliku se prikaže opozorilo. V istem dokumentu lahko vidimo tudi primer CSS (Cascading Style Sheets) kode, ki definira stil (izgled dokumenta prikazanega v spletnem brskalniku). CSS koda v primeru naslov (napisan v elementu <h1>)obarva v modro. Čeprav se največkrat uporablja za urejanje videza spletnih strani, napisanih v HTML kodi, ga lahko uporabljamo tudi za urejanje XML dokumentov. Glavni namen CSS-a je ločiti vsebino doku-

menta od predstavitve, kar se pri večjih projektih izkaže za zelo pomembno in omogoča tudi prikaz iste spletne strani na več načinov, v odvisnosti od naprave, ki do nje dostopa. V programu diplomske naloge lahko srečemo tudi uporabo knjižnice Jquery.js, katere namen je skrajšati stavke JavaScript in olajšati ostale funkcije, kot je npr. AJAX.

```
<html>
    <head>
        <title>Primer</title>
        <h1>Primer</h1>
        <style>h1{color:blue;}</style>
    </head>
    <body>
        <button id="zdravoSvet">Zdravo</button>
        <script>
                document.getElementById("zdravoSvet").onclick =
                function(){alert("Zdravo svet!");};
        </script>
     </body>
</html>
```

3.1 Knjižnica Three.js

Three.js je JavaScript knjižnica, ki jo uporabljamo za ustvarjanje in prikaz 3D računalniške grafike v spletnih brskalnikih[9]. Three.js uporablja knjižnico WebGL in jo naredi bolj lahkotno, na višjem nivoju. Potrebujemo veliko manj znanja in omogoča prikazovanje na HTML SVG ali HTML5 Canvas elementu (<canvas>,<svg>). Knjižnico je na portalu GitHub leta 2010 objavil Ricardo Cabello. Od takrat je doživela že mnogo sprememb. Eden izmed ciljev njenih izdelovalcev je bil, da njihov WebGL izrisovalnik premaga SVG ali Canvas izrisovalnik. Ponuja številne objektno sprogramirane elemente računalniške grafike.

Pomembni elementi so:

- animacije,
- zvoki,
- kamere (perspektivna, ortografksa ...),
- materiali (Lambertov, Phongov, globinski ...),
- pomnilniki (način shranjevanja različnih vrst geometrij),
- geometrije (škatla, cilinder, krog, izsek, ravnina ...),
- pomočniki (prikazovalnik osi, okostja ...),
- luči (ambientna svetloba-okoliška, usmerjena-svetilka, točkovna-žarnica ...),
- sence,
- nalagalniki (nalagalnik animacij, zvoka, podatkov geometrije, datotek ...),
- matematični objekti (ravnina, trikotnik, krogla, 3D vektor ...),
- interpolacije (linearna, kubična, diskretna ...),
- objekti (kost, skupina, črta, točka, 2D model, okostje, trikotnik, mreža točk iz trikotnikov ...),
- WebGL izrisovalnik,
- scena (v grafični svet dodajanje ali odstranjevanje objektov, nad katerimi lahko izvajamo skaliranje, tranzliranje in rotiranje),
- teksture,
- primeri (več kot 150 primerov programov, modeli, teksture, zvoki ...)
- ...

3.2 Knjižnica Prototype.js

Knjižnica Prototype.js je napisana v programskem jeziku JavaScript. Ustvaril jo je Sam Stephenson, in sicer februarja 2005[7]. Razvita je kot samostojna ali kot del večjih projektov (Ruby on Rails, Rico itd.). Ponuja razne funkcionalnosti, od okrajšav jezika JavaScript do funkcij kot je XMLHttpRequest. Najpomembnejši del te knjižnice predstavlja podpora razredov in razrednih objektov. To nam omogoča, da kodo lahko pišemo podobno, kot smo navajeni iz programskega jezika Java.
Primer:

```
var FirstClass = Class.create( {
    initialize: function () {
        this.data = "Hello World";
    }
});
```

3.3 Knjižnica Sylvester.js

Knjižnica Sylvester.js je namenjena uporabi vektorjev, matrik in geometrij[8] in lahko definiramo matrike in vektorje poljubnih dimenzij (knjižnica Three.js predvideva tri- ali štiridimenzionalne vektorje). Enako lahko definiramo neskončne linije in površine v 3D prostoru. Omogoča pisanje objektne kode, ki je prijazna za branje in stoji za matematiko, ki jo predstavlja.

3.4 Blender

Blender je brezplačen, odprtokoden program za ustvarjanje 3D modelov[1]. Podpira modeliranje, animiranje, simuliranje, izrisovanje, sestavljanje, sledenje gibanju, urejanje video posnetkov in celo ustvarjanje video iger. Program je bil uporabljen pri praktičnem delu izdelave aplikacije, za urejanje raznih 3D modelov oblačil. Tem lahko spreminjamo material, orientacijo in jih iz-

vozimo v željenem formatu za definiranje geometrije. Modeli oblačil, najdeni na spletu, so različnih formatov. Da ni bilo potrebno pisati kode za uvoz večih, so bili formati vseh oblačil spremenjeni v OBJ.

3.5 Format OBJ

Format OBJ je namenjen definiranju geometrije. Najprej so ga razvili v podjetju Wavefront Technologies in je zelo prijazen za branje. Poligonski 3D žičnati model, ki ga predstavljajo OBJ datoteke, je sestavljen iz trikotnikov ali štirikotnikov. OBJ format ima naslednje oznake:

- *v* x y z: vrstice, ki se začnejo s črko *v*, predstavljajo koordinate oglišč;
- *vt* u v: vrstice, ki se začnejo z *vt*, predstavljajo uv koordinate za mapiranje takstur;
- *vn* x y z: vrstice, ki se začnejo z *vn*, predstavljajo normale oglišč;
- *vp* u v w: vrstice, ki se začnejo z *vp*, predstavljajo oglišča iz parametrskega prostora;
- *f* indeks1 indeks2 indeks3: vrstice, ki se začnejo s *f*, predstavlja indekse oglišč, ki sestavljajo trikotnik (ali štirikotnik, če so štirje indeksi).

Primer dveh trikotnikov, ki skupaj predstavljata kvadrat:

```
v -1.000000 -1.000000 0.000000
v 1.000000 -1.000000 0.0000000
v 1.000000 1.000000 0.000000
v -1.000000 1.000000 0.000000
vt 0.000000 0.00000 0.000000
vt 1.000000 0.000000 0.000000
vt 1.000000 1.000000 0.000000
vt 0.000000 1.000000 0.000000
vn 0.000000 0.000000 1.000000
```

```
vn 0.000000 0.000000 1.000000
vn 0.000000 0.000000 1.000000
vn 0.000000 0.000000 1.000000
f 1/1/1 2/2/2 3/3/3
f 1/1/1 3/3/3 4/4/4
```

3.5.1 Format MTL

Z njim so opisani materiali modelov, ki jih definira format OBJ. Barve so podane v formatu RGB, kjer so vse vrednosti med 0 in 1. Osnovne lastnosti materiala so definirane z naslednjimi oznakami:

- newmtl Material: s to oznako definiramo nov material;
- K_a R G B: ambientna barva (svetloba, ki se odbija iz okolice);
- K_d R G B: difuzna barva (s to barvo je definirana glavna barva objekta);
- K_s R G B: spekularna barva (višje kot so vrednosti, bolj se material sveti);
- d *vrednost*: vrednost med 0 in 1 pove kakšna je transparentnost.

Primer rdečega materiala, ki se ne sveti:

```
newmtl Material_#35
Ka 1.000000 1.000000 1.000000
Kd 1.000000 0.000000 0.000000
Ks 0.000000 0.000000 0.000000
d 1.000000
```

Poglavje 4

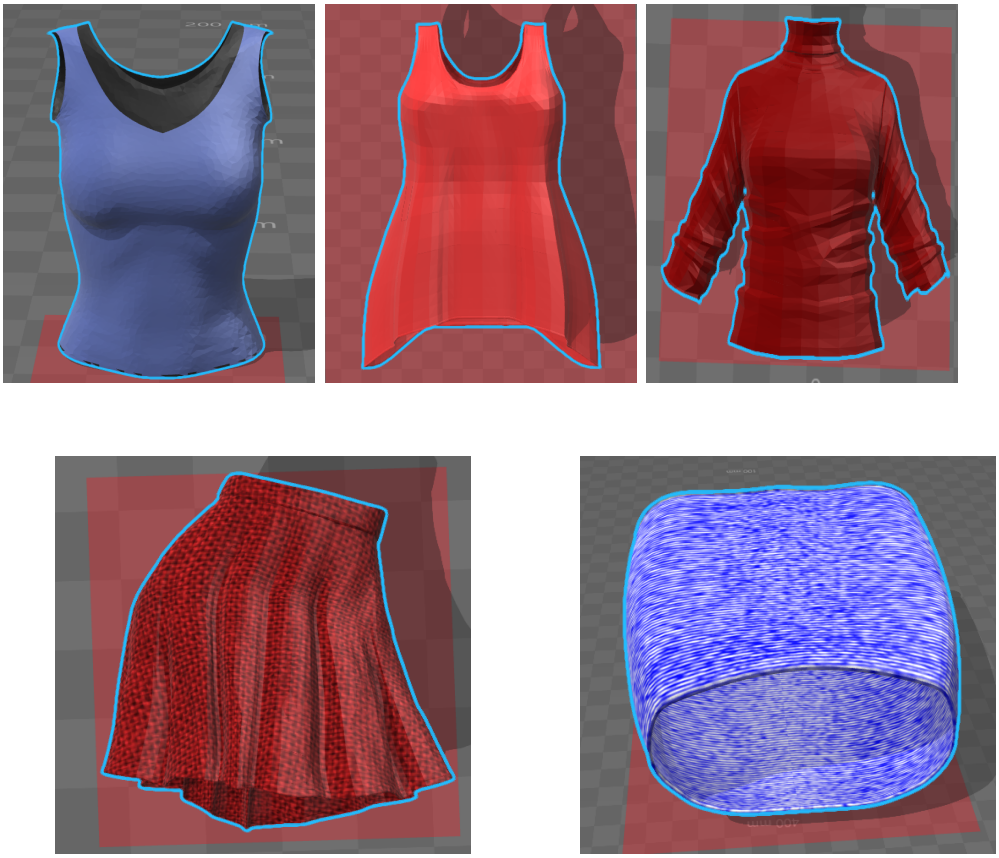
Praktični del

Cilj je razviti spletno stran, na kateri bo prikazano pomerjanje oblačil. Uporabniku mora biti na voljo 3D model človeka izbranega spola, ki mu lahko spreminja dimenzije. Prilagojen model lahko uporabimo za pomerjanje, ki je simulirano s pomočjo simulacije tkanin. Izdelana spletna stran je najprej skoraj prazna, nato pa se elementi dinamično dodajajo z uporabo raznih JavaScript metod. Najprej se izvede skripta, pridobljena s spletne strani Body Visualizer[2], ki izdela drsnike za spreminjanje dimenzij parametriziranega modela človeka. Vsak drsnik predstavlja en faktor skaliranja (raztezanja), ki se kasneje uporabi v programu. Ko so drsniki pripravljeni, se zažene prva metoda programa. Če iz izbranih elementov na spletni strani vidimo, da uporabnik hoče simulirati oblačila, potem najprej preberemo in shranimo izbran model obleke. Nato ga dodamo v sceno in njegov položaj prilagodimo za začetek simulacije. Sedaj naredimo nov objekt simuliranega oblačila, da lahko v simulaciji uporabljamo oz. spreminjamo geometrijo prebranih modelov. Potem v sceno dodamo vse elemente, ki jih potrebujemo. To so luči, kamera in parametriziran model človeka, ki ga tudi izdelamo. Nato nastopi simulacija tkanine. V simulaciji najprej simuliramo sile, nato raztezanje vzmeti med delci in na koncu stik med modelom človeka in tkanine. Želimo, da tkanina pada zaradi sile teže in se primerno ustavi na telesu. Po simulaciji točke simulirane tkanine prestavimo v prostor modela oblačila in

iteracijo končamo z izrisom na spletni strani. Če uporabnik noče simulirati oblačil in le spreminja model človeka, inicializiramo sceno in izrišemo model.

4.1 Uporabniška izkušnja

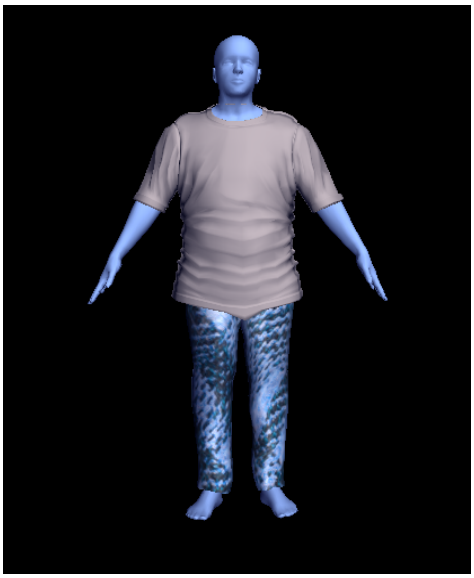
Ko uporabnik pride na spletno stran, se sreča z osnovnimi navodili, nato pa lahko izbere spol, ki ga želi v simulaciji. Sedaj vidi sedem drsnikov za spreminjanje dimenzij, ki predstavljajo višino, težo, obseg prsnega dela trupa, pasu, bokov, dolžino nog in število ur telovadbe na teden. Uporabnik lahko vnese željene mere, izbere pa lahko tudi oblačilo, ki ga želi pomeriti (slika 4.2 in 4.3).



Slika 4.2: Ženska oblačila.

Slika 4.3: Moška oblačila.

Ko izbere oblačilo, ima zanj na voljo več velikosti, razen za hlače, kjer je implementirano avtomatsko prilagajanje meram telesa. Odloči se lahko med mehkim ali trdim materialom. Trd material poskuša ohraniti originalen model in se težje prilagaja, mehak material pa dopušča večjo prilagodljivost, vendar tudi lažje pride do deformacij. Izbere lahko tudi svojo difuzno barvo materiala obleke, tako da vnese barvo v RGB formatu. Nato lahko zažene simulacijo.



Slika 4.4: Primer končane simulacije.

4.2 Implementacija

Ob kliku na gumb začetek simulacije, premikanju drsnikov ali prvem nalaganju spletne strani se izvede glavni program namenjen izrisovanju (metoda start). Ker je delovanje metod zelo gnezdeno in asinhrono, ob vsaki iteraciji najprej pobrišemo vrednosti spremenljivk in naredimo novo sceno. S tem se poskušamo izogniti napakam pomnilnika in procesorja.

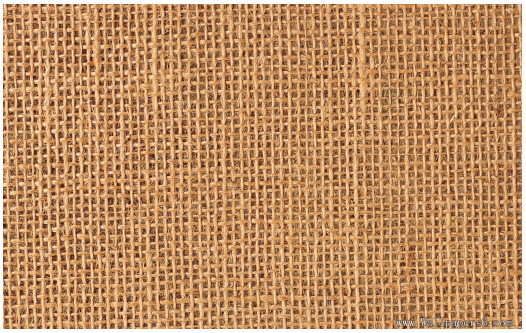
```
metoda start
  if izris_oblačil
      naloži_oblačila
      inicializiraj_simulacijo
      inicializiraj_sceno
      animiraj //kličemo simulacijo in izrisujemo sliko
  else
      inicializiraj_sceno
      animiraj //kličemo simulacijo in izrisujemo sliko
konec
```

4.2.1 Naloži oblačila

V tej metodi uporabljamo OBJLoader iz knjižnice *Three.js*, za nalaganje geometrije formata OBJ (geometrija oblačil) in MTLLoader za branje *.mtl* datotek, v katerih je opisan material modela, s katerim definiramo barvo površine. Nad prebranim objektom nato izvajamo transformacije premika (premikanje položaja v koordinatah sveta), raztega (spreminjane velikosti) in zasuka (obračanje modela). Z uporabo transformacij moramo model obleke poravnati z modelom človeka. Poleg izbranega oblačila preverimo tudi, kateri material je izbran (trd ali mehak) in ali je uporabnik vnesel svojo RGB vrednost za difuzno barvo materiala, v tem primeru jo spremenimo.

4.2.2 Inicializiraj simulacijo

V tej metodi za vsak izbran kos oblačila naredimo nov objekt razreda Cloth, ki predstavlja model mase in vzmeti, izdelan po geometriji oblačila. To dosežemo tako, da najprej iteriramo skozi vse točke izbranega modela in jim dodamo vektor položaja ($[x, y, z]$), prejšnega položaja, maso in pospešek (na začetku 0). Te podatke potrebujemo v simulaciji in so sedaj točke predstavljene kot delci. Nato izločimo tiste, ki imajo enake koordinate, da nimamo podvojenih, kar zelo pohitri simulacijo. Simulacija tkanine 2D mreže z enakomernimi razmiki med delci je zelo enostavna, vendar pri realnih modelih ni vse tako idealno, zato poskušamo narediti približek. Vpeljati moramo koordinato z. Če želimo razdeliti oblačilo po vseh x, y, z koordinatah, je iteriranja preveč, saj se težavnost z gnezdenjem zank veča eksponentno. Zato poiščemo sredinsko z koordinato oblačila in model razdelimo na dva dela (delce, ki imajo koordinato z manjšo in ostale). Od tod naprej oblačilo simuliramo kot 2D mrežo, vendar imamo še vedno neenakomerne razdalje med delci. Oba dela tkanine razdelimo po x intervalih, da dobimo niti delcev, kot jih lahko vidimo na pravem oblačilu (slika 5.1). Nato x intervale razdelimo



Slika 4.5: Primer niti tkanine.

po y intervalih in vzmeti lahko začnemo postavljati med delce. Iteriramo skozi intervale in postavimo vse tri vrste vzmeti. Najprej strukturne vzmeti postavimo čez vse delce na istem x, y intervalu, saj jih smatramo kot en delec. Preverimo, kateri delec je v naslednjem x intervalu in ju povežemo; enako na-

redimo v smeri y. Tukaj moramo biti pozorni na obliko oblačil (slika 4.6), saj

Slika 4.6: Primer oblike majice.

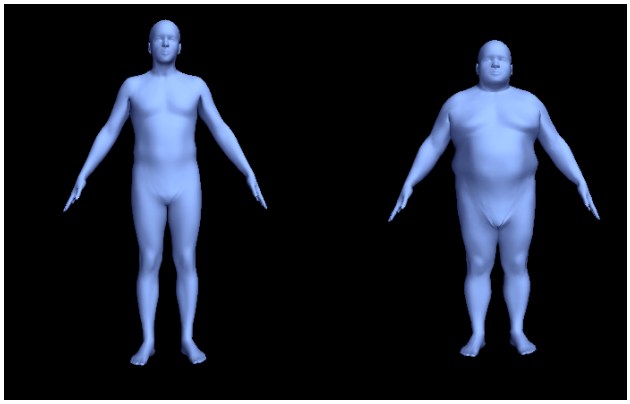
vzmeti nočemo postaviti med spodnji del rokava in trup majice. To bi otežilo prilagajanje rokava roki modela človeka, zato moramo preverjati razdaljo med delci, da vzmeti niso predolge. Če smo uspešno postavili strukturno vzmet v smeri x in y, lahko tudi strižno vzmet postavimo v diagonalni smeri (če razdalja ni prevelika), na enak način postavimo upogibne vzmeti. Če ustvarjamo model hlač poiščemo najvišji delec in shranimo vse, ki so oddaljeni v smeri y za manj kot višino pasu (poljubna konstanta). Sledi vezanje obeh površin obleke s pomočjo vzmeti. Največja dolžina teh vzmeti je izračunana po izrazu $povp/2$, če je izbran trd material, ali po izrazu $povp/5$, če je izbran mehak material, kjer je spremenljivka $povp$ povprečna oddaljenost delcev od prve površine do druge.

4.2.3 Inicializiraj sceno

V tej metodi urejamo sceno in vanjo dodajamo elemente, ki jih potrebujemo. Dodamo perspektivno kamero, orbitni kontroler (namenjen obračanju in premikanju kamere), ambientno luč (predstavlja svetlobo iz okolice), tri usmerjene luči (predstavljajo svetilko), parametriziran model človeka in vsem elementom določimo tudi položaj v sceni. Če izrisujemo oblačila (nastopila bo simulacija), modelu človeka obrišemo škatlo, ki jo razdelimo na več manjših, nato preverimo katere točke modela človeka se nahajajo v posamezni škatli in reference na te točke shranimo v tabele (vsaka tabela predstavlja eno škatlo

točk). Detekcija trkov velja za ozko grlo simulacije tkanin, zato jo poskušamo kar se da poenostaviti. Največje škatle nočemo razdeliti na preveč manjših, saj bi bilo tako težje najti del telesa, ki je najbližji delu tkanine.

4.2.4 Parametriziran model človeka



Slika 4.7: Spreminjane modela človeka s pomočjo odmikov

Kako spreminjamo model človeka? Najprej imamo navaden 3D model človeka, ki velja za "povprečnega" in se prikaže ob prvem nalaganju modela. Da lahko spremenimo geometrijo modela, imamo za vsako mero geometrijo, ki definira odmik, za točno določen del telesa. V našem primeru torej omogočamo spreminjanje modela glede na višino, težo, obseg prsnega dela trupa, pasu, bokov, dolžine nog in število ur telovadbe na teden. Za vsako mero imamo tudi svoj faktor skaliranja, ki je na začetku nevtralen in na model ne vpliva. Faktorji so predstavljeni z drsniki, ki jih uporabnik lahko premika iz začetne pozicije in jih s tem zmanjšuje ali veča. Za spreminjanje faktorjev glede na vrednosti drsnikov je uporabljena skripta conditional_gaussian.js, pridobljena s spletne strani Body Visualizer[2]. Najprej imamo shranjene točke, normale in indekse geometrije povprečnega modela. Indeksi nam povedo, katere točke skupaj tvorijo trikotnike, iz katerih je sestavljen model. Če prvi trikotnik tvorijo prvi, drugi in tretji indeks, drugi trikotnik tvorijo četrti, peti in šesti indeks. To ni zelo učinkovito, saj bi se indeksi lahko

ponavljali (npr. podatkovni model pahljače). Shranjene imamo tudi faktorje skaliranja. Najprej vse faktorje seštejemo in dobimo vsoto. Položaje vseh začetnih točk (x, y, z) pomnožimo z izrazom $(1 - vsotaFaktorjev)$. Nato se sprehodimo čez vse faktorje skaliranja, v našem primeru čez vseh 7, in za vsakega vzamemo vse točke in normale modela odmikov. Potem se v notranji zanki sprehodimo čez vse položaje točk glavnega modela, ki smo jih prej množili, in jim prištejemo produkt položajev modela odmika (imajo enak indeks) s faktorjem skaliranja, ki mu ta model pripada. Enako naredimo z normalami, le da jih prej ne množimo z nobenim izrazom. V primeru, da je model človeka bolj kompleksen, deformacije normal nočemo izvajati v isto smer, kot so že usmerjene. Sedaj so točke, normale in indeksi pripravljeni, da izrišemo glede na faktorje skaliranja spremenjen model človeka.

4.2.5 Animiraj

To je asinhrona metoda, ki se zaradi prikazovanja kliče znova in znova, zato jo moramo po izvajanju ustaviti, saj nočemo, da ob več klicih programa teče več paralelnih metod hkrati (zmanjka rezerviranega pomnilnika in program preneha delovati). Ta metoda zaprosi za okvir slike (angl. frame) in pokliče simulacijo tkanine. Nato sledi izrisovanje. Tu se posodobijo točke modela oblačila glede na položaje delcev, ki se spreminjajo v simulaciji. Iteracija se konča z izrisom scene na zaslon.

4.2.6 Simulacija

Koraki simulacije:

- simulacija sil na tkanino,
- simulacija raztezanja vzmeti,
- simulacija stika človeka in tkanine.

4.2.7 Simulacija sil na tkanino

Ker je tema diplomske naloge pomerjanje oblačil v garderobi, je edina sila, ki jo simuliramo gravitacija (predvidevamo, da ni vetra). Iteriramo skozi vse unikatne delce in za vsakega izračunamo pospešek. To storimo z uporabo Newtonovega drugega zakona (enačba 2.1). Nov položaj delca izračunamo s poenostavljeno Verletovo integracijo. Tu upoštevamo tudi faktor blaženja (enačba 2.5), ki ga definiramo kot konstanto (manjšo od 1). Koraki računanja so naslednji:

1.
$$V_r = V_t - V_p \tag{4.1}$$

2.
$$V_u = (V_r * k_b) + V_t \tag{4.2}$$

3.
$$s = a * t^2 \tag{4.3}$$
$$V_n = V_u + V_s \tag{4.4}$$

V 1. koraku odštejemo vektor prejšnega položaja V_p od trenutnega položaja V_t in dobimo vektor razlike položajev V_r. V 2. koraku vektor razlike skalarno množimo s faktorjem blaženja k_b (manjši od 1) in prištejemo vektor trenutnega položaja. Dobimo vektor premika V_u, utežen s faktorjem blaženja. V 3. koraku enačba 4.3 predstavlja fizikalni zakon, kjer je pot definirana kot produkt pospeška in kvadrata časa. V tej točki simulacije sta pospešek in časovni interval znana, zato je izračun opravljene poti enostaven. Nov položaj delca (vektor V_n) dobimo tako, da uteženemu premiku prištejemo vektor opravljene poti V_s.

4.2.8 Simulacija raztezanja vzmeti

Cilj metode je obnoviti originalno dolžino vzmeti, vendar ne v enem koraku. To storimo tako, da izračunamo vektor razlike (položaja delcev odštejemo)

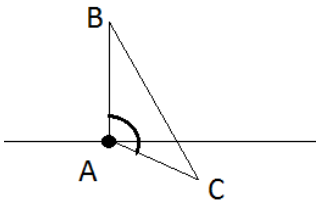
med delcema na vsaki strani vzmeti in vektor popravljanja.

$$V_p = (V_r * (1 - d/d_t)) * 0.5 \tag{4.5}$$

Enačba 4.5 pravi: Vektor popravljanja je skalarni produkt vektorja razlike z izrazom $(1-dolzinaVzmeti/trenutnaDolzinaVzmeti)*0.5$. Razliko utežimo z razmerjem med originalno dolžino vzmeti in trenutno dolžino, nato jo razpolovimo, da lahko izvajamo popravljanje v več korakih. Dobljeni vektor prištejemo položaju delca na prvi strani vzmeti in odštejemo od položaja delca na drugi strani. Tako vzmet postopoma vračamo proti začetni dolžini in večkrat kot iteriramo to proceduro, hitreje se vzmet vrača v prvotno stanje.

4.2.9 Simulacija stika človeka in tkanine

Zopet se sprehajamo skozi vse unikatne delce in za vsakega posebej se sprehodimo skozi vse škatle, ki smo jih definirali ob inicializaciji scene. S primerjanjem x, y, z koordinat (poznamo koordinate začetne in končne točke telesne diagonale škatle), poiščemo škatlo, v kateri se delec nahaja. Tako ni treba preverjati trkov z vsemi deli telesa. Poiščemo najbližjo točko telesa (najkrajša razdalja med delcem tkanine in točko telesa), kopiji te točke pa prištejemo normalo (vektor, definiran v geometriji telesa) z enakim indeksom, kot ga ima točka. Izračunamo razdaljo med točko telesa, točko normale in delcem tkanine in tako dobimo trikotnik. Na sliki 4.8 je prikazan idealen



Slika 4.8: Primer trikotnika razdalj.

primer. Oglišče A predstavlja točko telesa, oglišče B točko normale in oglišče

C delec tkanine, ki je pod kožo modela človeka. Kosinusni izrek (enačba 4.6) nam omogoča, da izračunamo kot, označen na sliki 4.8. Če je kot večji od 90 stopinj in je oddaljenost delca dovolj majhna, takrat mu prištejemo normalo (sklepamo, da je delec pod kožo in ga premaknemo proti površju). Ker koža ni vedno ravna, včasih s kotom 90 stopinj tkanine ne potisnemo dovolj daleč (poskušamo izboljšati sliko), zato preverimo, če je kot večji od 45 stopinj in razdalja med delcem in kožo zelo majhna. V tem primeru delec tkanine potisnemo za manjšo velikost normale (napaka potiskanja je večja). V simulaciji obravnavamo tudi delce, ki so namenjeni pasu hlač. Za vsak takšen delec zopet poiščemo najbližjo točko telesa in mu prištejemo obrnjen vektor normale. Tako jih premaknemo proti telesu in jim ne dovolimo padati z gravitacijo. Na koncu simulacije izenačimo položaje neunikatnih delcev z unikatnimi.

$$a^2 = b^2 + c^2 - 2bc\cos\alpha \tag{4.6}$$

Poglavje 5

Zaključek

V delu smo obnovili osnovne metode simulacije tkanin. Implementirali smo metodo mase in vzmeti, ki predstavlja najnovejšo. Videli smo, da je tkanina v metodi predstavljena kot mreža delcev, ki so povezani z vzmetmi. Predstavili smo pojem 3D modelov in videli, da so objekti v grafičnem svetu le množica obarvanih likov. Parametrizirani so tisti, ki jim z raznimi funkcijami spreminjamo položaje točk, npr. večanje polmera rezinam modela. Ogledali smo si osnove spletnega razvoja in poudarili, da je JavaScript zelo dinamičen programski jezik. Spoznali smo knjižnico Prototype.js (največkrat uporabljena za izdelavo razredov), Sylvester.js (namenjena računanju z matrikami in vektorji) in Three.js, ki nam ponuja mnogo koristnih elementov, od hierarhične scene do geometrij. Na kratko smo predstavili format zapisa geometrije 3D modelov OBJ, ki je eden najenostavnejših, način, kako razberemo osnovne lastnosti njihovega materiala (zapisanega v formatu MTL) in urejevalnik modelov Blender. V praktičnem delu smo povedali, da izdelan program omogoča spreminjanje 3D modela človeka, prilagajanje modelov oblačil, pomerjanje oblačil s pomočjo simulacije tkanin in predstavili njegovo delovanje.

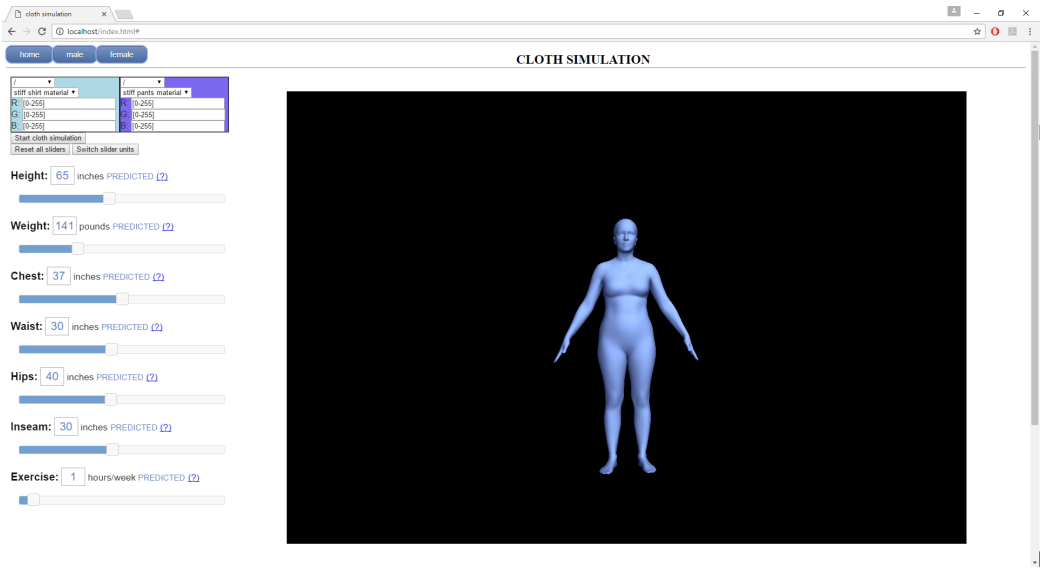
5.1 Rezultati in problemi

Nastala je spletna stran, na kateri lahko uporabniki pomerjajo oblačila. Izberejo lahko spol, sedem telesnih mer, oblačilo, velikost in barvo. Odločijo se lahko med mehkim in trdim materialom oblačila, kar aplikacijo naredi bolj fleksibilno. Kadarkoli lahko simulacijo tudi zaženejo in si ogledajo, kako se oblačila prilagodijo modelu telesa. Simulacija teče gladko, če model oblačila ustrezno poravnamo z modelom človeka. Manjka ji nekaj realizma, saj konstante, kot so masa delcev oblačil, upor zraka in gravitacija, niso realne. Nekaj težav je predstavljala asinhronost nekaterih metod, saj so se hotele izvajati vzporedno, kar je preprečilo delovanje programa. Pridobivanje modelov oblačil se je prav tako izkazalo za težavno, saj pri simulaciji oblačil nočemo imeti velikega števila točk, drugače je izvajanje prepočasno in modeli oblačil niso optimizirani za simulacijo tkanin. Veliko časa je bilo vloženega v transformacije velikosti in položaja modelov oblačil po branju, saj želimo, da ima vsak model človeka na voljo velikost oblačila, ki se mu uspešno prilagodi. Pri aplikacijah pomerjanja oblačil avtomatsko prilagajanje velikosti ni smiselno, saj uporabniku želimo ponuditi določene velikosti (kot v resničnem življenju).

5.2 Priporočila

Razvili bi lahko različne materiale tkanine, ki vzmeti naredijo poljubno močne in simulirali realne. Če ne razvijamo aplikacije za pomerjanje oblačil (oblačila imajo fiksno velikost), ampak želimo simulacijo tkanin vpeljati npr. v računalniško igro, priporočamo, da model človeka pred začetkom simulacije avtomatsko ovijemo s točkami modela oblačila. Tako se izognemo ročnemu prilagajanju transformacij različnih modelov oblačil. Ko inicializiramo simulacijo, model oblačila razdelimo na dva dela. S tem naredimo le približek pravemu oblačilu. Rezanje na več delov bi bilo smiselno, saj je metoda mase in vzmeti definirana za 2D tkanino. Detekcija trkov med modelom človeka in oblačila bi bila lahko še optimizirana. Poiskali bi lahko optimalno število

škatelj ali raziskali druge metode. Ob detekciji trka, potiskanje v smeri vektorja normale (kot je implementirano trenutno) ne deluje v 100% primerov, zato so raziskave smiselne.



Slika 5.1: Primer končne spletne strani.

Literatura

[1] Blender reference manual. Dosegljivo: https://docs.blender.org/manual/en/dev/.

[2] Body builde. Dosegljivo: http://bodyvisualizer.com/.

[3] Cloth modifier — max 3d — autodesk knowledge network. Dosegljivo: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2017/ENU/3DSMax/files/GUID-1663898B-7C53-4123-9D54-7B7EE843FB92-htm.html.

[4] Create and edit ncloth — maya — autodesk knowledge network. Dosegljivo: https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Maya/files/GUID-6A645A7C-1583-476D-86EA-B56E317B8D6F-htm.html.

[5] Javascript. Dosegljivo: https://www.javascript.com/.

[6] Makehuman — open source tool for making 3d characters. Dosegljivo: http://www.makehuman.org/.

[7] Prototype javascript framework: a foundation for ambitious web applications. Dosegljivo: http://prototypejs.org/.

[8] Sylvester. Dosegljivo: http://sylvester.jcoglan.com/.

[9] three.js - javascript 3d library. Dosegljivo: https://threejs.org/.

[10] Ustvarjanje 3d-modelov in likov — adobe fuse cc (predogled). Dosegljivo: http://www.adobe.com/si/products/fuse.html.

[11] W3schools online web tutorials. Dosegljivo: https://www.w3schools.com/.

[12] David E. Breen, Donald H. House, and Phillip H. Getto. A physically-based particle model of woven cloth. *The Visual Computer*, 8(5):264–277, 1992.

[13] David R Haumann and Richard E Parent. The behavioral test-bed: Obtaining complex behavior from simple rules. *The Visual Computer*, 4(6):332–347, 1988.

[14] Donald H House and David E Breen. *Cloth modeling and animation*. AK Peters, Ltd., 2000.

[15] Andrew Hoyer. The cloth simulation. Dosegljivo: http://andrew-hoyer.com/experiments/cloth/.

[16] Mustafa Kasap and Nadia Magnenat-Thalmann. Parameterized human body model for real-time applications. In *Cyberworlds, 2007. CW'07. International Conference on*, pages 160–167. IEEE, 2007.

[17] Xavier Provot et al. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, pages 147–147. Canadian Information Processing Society, 1995.

[18] Jerry Weil. The synthesis of cloth objects. *SIGGRAPH Comput. Graph.*, 20(4):49–54, August 1986.